

RRRRRRRR	PPPPPPPP	GGGGGGGG	PPPPPPPP	RRRRRRRR	IIIIII	NN	NN	TTTTTTTT				
RRRRRRRR	PPPPPPPP	GGGGGGGG	PPPPPPPP	RRRRRRRR	IIIIII	NN	NN	TTTTTTTT				
RR	RR	PP	PP	GG	PP	PP	RR	RR	II	NN	NN	TT
RR	RR	PP	PP	GG	PP	PP	RR	RR	II	NN	NN	TT
RR	RR	PP	PP	GG	PP	PP	RR	RR	II	NNNN	NN	TT
RR	RR	PP	PP	GG	PP	PP	RR	RR	II	NNNN	NN	TT
RRRRRRRR	PPPPPPPP	GG	PPPPPPPP	RRRRRRRR	II	NN	NN	TT				
RRRRRRRR	PPPPPPPP	GG	PPPPPPPP	RRRRRRRR	II	NN	NN	TT				
RR	RR	PP	GG	GGGGGG	PP	RR	RR	II	NN	NNNN	TT	
RR	RR	PP	GG	GGGGGG	PP	RR	RR	II	NN	NNNN	TT	
RR	RR	PP	GG	GG	PP	RR	RR	II	NN	NN	TT	
RR	RR	PP	GG	GG	PP	RR	RR	II	NN	NN	TT	
RR	RR	PP	GGGGGG	PP	RR	RR	IIIIII	NN	NN	NN	TT	
RR	RR	PP	GGGGGG	PP	RR	RR	IIIIII	NN	NN	NN	TT	

```
1 0001 0 MODULE RPG$PRINT( *TITLE,'Support output to RPG PRINTER files'
2 0002 0           IDENT = '1-003'
3 0003 0           ) =
4 0004 1 BEGIN
5 0005 1
6 0006 1 ***** ****
7 0007 1 *
8 0008 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
9 0009 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
10 0010 1 * ALL RIGHTS RESERVED.
11 0011 1 *
12 0012 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
13 0013 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
14 0014 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
15 0015 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
16 0016 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
17 0017 1 * TRANSFERRED.
18 0018 1 *
19 0019 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
20 0020 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
21 0021 1 * CORPORATION.
22 0022 1 *
23 0023 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
24 0024 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
25 0025 1 *
26 0026 1 *
27 0027 1 ****
28 0028 1
29 0029 1
30 0030 1 ++
31 0031 1
32 0032 1 FACILITY: RPGII SUPPORT
33 0033 1
34 0034 1 ABSTRACT:
35 0035 1
36 0036 1 This module contains the RTL routines to handle output to RPG
37 0037 1 PRINTER files for VAX-11 RPGII.
38 0038 1
39 0039 1 ENVIRONMENT: VAX/VMS user mode
40 0040 1
41 0041 1 AUTHOR: Debess Grabazs, CREATION DATE: 20-December-1982
42 0042 1
43 0043 1 MODIFIED BY:
44 0044 1
45 0045 1 1-001 - Original version. DG 20-DEC-82
46 0046 1 1-002 - Added code review comments - most notable code change is the different
47 0047 1 looping techniques for first page forms positioning. DG 26-MAY-83
48 0048 1 1-003 - 1. Add support for overprinting lines. DJB 27-Jun-1983
49 0049 1 2. Only turn on overflow indicator if space or skip after is passed
50 0050 1 the overflow line. LPT 5-Jul-1983
51 0051 1
52 0052 1 --
53 0053 1 <BLF/PAGE>
```

```
55      0054 1  XSBTTL 'Declarations'  
56      0055 1  +  
57      0056 1  |  PROLOGUE FILE:  
58      0057 1  |-  
59      0058 1  |  
60      0059 1  REQUIRE 'RTLIN:RPGPROLOG';           ! Switches, PSECTs, macros,  
61      0124 1  |  
62      0125 1  |  
63      0126 1  |  +  
64      0127 1  |  |  LINKAGES  
65      0128 1  |  |  NONE  
66      0129 1  |  |-  
67      0130 1  |  
68      0131 1  |  +  
69      0132 1  |  |  TABLE OF CONTENTS:  
70      0133 1  |  |-  
71      0134 1  |  
72      0135 1  FORWARD ROUTINE  
73      0136 1  RPG$PRINT,  
74      0137 1  RPGTERM_PRINT;  
75      0138 1  |  
76      0139 1  |  +  
77      0140 1  |  |  INCLUDE FILES  
78      0141 1  |  |  NONE  
79      0142 1  |  |-  
80      0143 1  |  
81      0144 1  |  +  
82      0145 1  |  |  MACROS  
83      0146 1  |  |-  
84      0147 1  |  
85      0148 1  MACRO          PREFIX = 0,0,8,0%.  
86      0149 1  |  
87      0150 1  |  POSTFIX = 0,8,8,0%;           ! Record header block fields  
88      0151 1  |  
89      0152 1  |  +  
90      0153 1  |  |  EQUATED SYMBOLS  
91      0154 1  |  |  NONE  
92      0155 1  |  |-  
93      0156 1  |  
94      0157 1  |  +  
95      0158 1  |  |  EXTERNAL REFERENCES  
96      0159 1  |  |-  
97      0160 1  |  
98      0161 1  EXTERNAL ROUTINE  
99      0162 1  LIB$GET COMMAND,  
100     0163 1  STR$UPCASE;           ! Get line from SYSSCOMMAND  
101     0164 1  |           ! Convert string to uppercase  
102     0165 1  EXTERNAL LITERAL  
103     0166 1  RPG$_EXTINDOFF;      ! File not open error  
104     0167 1
```

```

106 0168 1 %SBTTL 'RPG$PRINT - Support output to RPG PRINTER files'
107 0169 1 GLOBAL ROUTINE RPG$PRINT(
108 0170 1 RAB: REF $RAB_DECL ! RAB of file to be printed
109 0171 1 ) =
110 0172 1 ++
111 0173 1
112 0174 1
113 0175 1 FUNCTIONAL DESCRIPTION:
114 0176 1
115 0177 1 This routine supports output to RPG PRINTER files. It is called by
116 0178 1 the compiled code once for each write to a PRINTER file.
117 0179 1
118 0180 1 The main function of this routine is to fill in the two-byte
119 0181 1 fixed-length control area which is associated with each record
120 0182 1 and to write the print record to the file. This control area
121 0183 1 contains the spacing controls for a print record. If spacing
122 0184 1 and skipping are both specified for the same line, they are
123 0185 1 performed in the following sequence:
124 0186 1   o Skip before
125 0187 1   o Space before
126 0188 1   o Print the line
127 0189 1   o Skip after
128 0190 1   o Space after.
129 0191 1 The secondary function of this routine is to detect page
130 0192 1 overflow. This occurs only the first time one of the following
131 0193 1 conditions occurs on the current page:
132 0194 1   o A line is printed on the overflow line
133 0195 1   o A line is printed past the overflow line
134 0196 1   o The overflow line is passed during a space operation
135 0197 1   o The overflow line is passed during a skip operation
136 0198 1   (to a line on the current page).
137 0199 1 A special function of this routine is to allow "first page"
138 0200 1 forms positioning. If both RPG$V_CTX_1PFORMS and RPG$V_CTX_FIRST
139 0201 1 are set on, this routine will do the following:
140 0202 1   o PUT the record
141 0203 1   o If RMS returns a failure status, return
142 0204 1   o Issue a message to SYSSCOMMAND to ask the user whether
143 0205 1   forms are positioned correctly
144 0206 1   o Accept "continue" or "retry" as a response
145 0207 1   o If the user responds with "retry", go back to step 1
146 0208 1   o If the user responds with "continue", clear
147 0209 1   RPG$V_CTX_FIRST and return.
148 0210 1
149 0211 1 CALLING SEQUENCE:
150 0212 1
151 0213 1   return_status.wlc.v = RPG$PRINT (rab.rr.r)
152 0214 1
153 0215 1 FORMAL PARAMETERS:
154 0216 1
155 0217 1   rab
156 0218 1   address of the RAB of the file to be
157 0219 1   printed.
158 0220 1
159 0221 1
160 0222 1
161 0223 1
162 0224 1
163 0225 1
164 0226 1
165 0227 1
166 0228 1
167 0229 1
168 0230 1
169 0231 1
170 0232 1
171 0233 1
172 0234 1
173 0235 1
174 0236 1
175 0237 1
176 0238 1
177 0239 1
178 0240 1
179 0241 1
180 0242 1
181 0243 1
182 0244 1
183 0245 1
184 0246 1
185 0247 1
186 0248 1
187 0249 1
188 0250 1
189 0251 1
190 0252 1
191 0253 1
192 0254 1
193 0255 1
194 0256 1
195 0257 1
196 0258 1
197 0259 1
198 0260 1
199 0261 1
200 0262 1
201 0263 1
202 0264 1
203 0265 1
204 0266 1
205 0267 1
206 0268 1
207 0269 1
208 0270 1
209 0271 1
210 0272 1
211 0273 1
212 0274 1
213 0275 1
214 0276 1
215 0277 1
216 0278 1
217 0279 1
218 0280 1
219 0281 1
220 0282 1
221 0283 1
222 0284 1
223 0285 1
224 0286 1
225 0287 1
226 0288 1
227 0289 1
228 0290 1
229 0291 1
230 0292 1
231 0293 1
232 0294 1
233 0295 1
234 0296 1
235 0297 1
236 0298 1
237 0299 1
238 0200 1
239 0201 1
240 0202 1
241 0203 1
242 0204 1
243 0205 1
244 0206 1
245 0207 1
246 0208 1
247 0209 1
248 0210 1
249 0211 1
250 0212 1
251 0213 1
252 0214 1
253 0215 1
254 0216 1
255 0217 1
256 0218 1
257 0219 1
258 0220 1
259 0221 1
260 0222 1
261 0223 1
262 0224 1
263 0225 1
264 0226 1
265 0227 1
266 0228 1
267 0229 1
268 0220 1
269 0221 1
270 0222 1
271 0223 1
272 0224 1
273 0225 1
274 0226 1
275 0227 1
276 0228 1
277 0229 1
278 0220 1
279 0221 1
280 0222 1
281 0223 1
282 0224 1
283 0225 1
284 0226 1
285 0227 1
286 0228 1
287 0229 1
288 0220 1
289 0221 1
290 0222 1
291 0223 1
292 0224 1
293 0225 1
294 0226 1
295 0227 1
296 0228 1
297 0229 1
298 0220 1
299 0221 1
300 0222 1
301 0223 1
302 0224 1
303 0225 1
304 0226 1
305 0227 1
306 0228 1
307 0229 1
308 0220 1
309 0221 1
310 0222 1
311 0223 1
312 0224 1
313 0225 1
314 0226 1
315 0227 1
316 0228 1
317 0229 1
318 0220 1
319 0221 1
320 0222 1
321 0223 1
322 0224 1
323 0225 1
324 0226 1
325 0227 1
326 0228 1
327 0229 1
328 0220 1
329 0221 1
330 0222 1
331 0223 1
332 0224 1
333 0225 1
334 0226 1
335 0227 1
336 0228 1
337 0229 1
338 0220 1
339 0221 1
340 0222 1
341 0223 1
342 0224 1
343 0225 1
344 0226 1
345 0227 1
346 0228 1
347 0229 1
348 0220 1
349 0221 1
350 0222 1
351 0223 1
352 0224 1
353 0225 1
354 0226 1
355 0227 1
356 0228 1
357 0229 1
358 0220 1
359 0221 1
360 0222 1
361 0223 1
362 0224 1
363 0225 1
364 0226 1
365 0227 1
366 0228 1
367 0229 1
368 0220 1
369 0221 1
370 0222 1
371 0223 1
372 0224 1
373 0225 1
374 0226 1
375 0227 1
376 0228 1
377 0229 1
378 0220 1
379 0221 1
380 0222 1
381 0223 1
382 0224 1
383 0225 1
384 0226 1
385 0227 1
386 0228 1
387 0229 1
388 0220 1
389 0221 1
390 0222 1
391 0223 1
392 0224 1
393 0225 1
394 0226 1
395 0227 1
396 0228 1
397 0229 1
398 0220 1
399 0221 1
400 0222 1
401 0223 1
402 0224 1
403 0225 1
404 0226 1
405 0227 1
406 0228 1
407 0229 1
408 0220 1
409 0221 1
410 0222 1
411 0223 1
412 0224 1
413 0225 1
414 0226 1
415 0227 1
416 0228 1
417 0229 1
418 0220 1
419 0221 1
420 0222 1
421 0223 1
422 0224 1
423 0225 1
424 0226 1
425 0227 1
426 0228 1
427 0229 1
428 0220 1
429 0221 1
430 0222 1
431 0223 1
432 0224 1
433 0225 1
434 0226 1
435 0227 1
436 0228 1
437 0229 1
438 0220 1
439 0221 1
440 0222 1
441 0223 1
442 0224 1
443 0225 1
444 0226 1
445 0227 1
446 0228 1
447 0229 1
448 0220 1
449 0221 1
450 0222 1
451 0223 1
452 0224 1
453 0225 1
454 0226 1
455 0227 1
456 0228 1
457 0229 1
458 0220 1
459 0221 1
460 0222 1
461 0223 1
462 0224 1
463 0225 1
464 0226 1
465 0227 1
466 0228 1
467 0229 1
468 0220 1
469 0221 1
470 0222 1
471 0223 1
472 0224 1
473 0225 1
474 0226 1
475 0227 1
476 0228 1
477 0229 1
478 0220 1
479 0221 1
480 0222 1
481 0223 1
482 0224 1
483 0225 1
484 0226 1
485 0227 1
486 0228 1
487 0229 1
488 0220 1
489 0221 1
490 0222 1
491 0223 1
492 0224 1
493 0225 1
494 0226 1
495 0227 1
496 0228 1
497 0229 1
498 0220 1
499 0221 1
500 0222 1
501 0223 1
502 0224 1
503 0225 1
504 0226 1
505 0227 1
506 0228 1
507 0229 1
508 0220 1
509 0221 1
510 0222 1
511 0223 1
512 0224 1
513 0225 1
514 0226 1
515 0227 1
516 0228 1
517 0229 1
518 0220 1
519 0221 1
520 0222 1
521 0223 1
522 0224 1
523 0225 1
524 0226 1
525 0227 1
526 0228 1
527 0229 1
528 0220 1
529 0221 1
530 0222 1
531 0223 1
532 0224 1
533 0225 1
534 0226 1
535 0227 1
536 0228 1
537 0229 1
538 0220 1
539 0221 1
540 0222 1
541 0223 1
542 0224 1
543 0225 1
544 0226 1
545 0227 1
546 0228 1
547 0229 1
548 0220 1
549 0221 1
550 0222 1
551 0223 1
552 0224 1
553 0225 1
554 0226 1
555 0227 1
556 0228 1
557 0229 1
558 0220 1
559 0221 1
560 0222 1
561 0223 1
562 0224 1
563 0225 1
564 0226 1
565 0227 1
566 0228 1
567 0229 1
568 0220 1
569 0221 1
570 0222 1
571 0223 1
572 0224 1
573 0225 1
574 0226 1
575 0227 1
576 0228 1
577 0229 1
578 0220 1
579 0221 1
580 0222 1
581 0223 1
582 0224 1
583 0225 1
584 0226 1
585 0227 1
586 0228 1
587 0229 1
588 0220 1
589 0221 1
590 0222 1
591 0223 1
592 0224 1
593 0225 1
594 0226 1
595 0227 1
596 0228 1
597 0229 1
598 0220 1
599 0221 1
600 0222 1
601 0223 1
602 0224 1
603 0225 1
604 0226 1
605 0227 1
606 0228 1
607 0229 1
608 0220 1
609 0221 1
610 0222 1
611 0223 1
612 0224 1
613 0225 1
614 0226 1
615 0227 1
616 0228 1
617 0229 1
618 0220 1
619 0221 1
620 0222 1
621 0223 1
622 0224 1
623 0225 1
624 0226 1
625 0227 1
626 0228 1
627 0229 1
628 0220 1
629 0221 1
630 0222 1
631 0223 1
632 0224 1
633 0225 1
634 0226 1
635 0227 1
636 0228 1
637 0229 1
638 0220 1
639 0221 1
640 0222 1
641 0223 1
642 0224 1
643 0225 1
644 0226 1
645 0227 1
646 0228 1
647 0229 1
648 0220 1
649 0221 1
650 0222 1
651 0223 1
652 0224 1
653 0225 1
654 0226 1
655 0227 1
656 0228 1
657 0229 1
658 0220 1
659 0221 1
660 0222 1
661 0223 1
662 0224 1
663 0225 1
664 0226 1
665 0227 1
666 0228 1
667 0229 1
668 0220 1
669 0221 1
670 0222 1
671 0223 1
672 0224 1
673 0225 1
674 0226 1
675 0227 1
676 0228 1
677 0229 1
678 0220 1
679 0221 1
680 0222 1
681 0223 1
682 0224 1
683 0225 1
684 0226 1
685 0227 1
686 0228 1
687 0229 1
688 0220 1
689 0221 1
690 0222 1
691 0223 1
692 0224 1
693 0225 1
694 0226 1
695 0227 1
696 0228 1
697 0229 1
698 0220 1
699 0221 1
700 0222 1
701 0223 1
702 0224 1
703 0225 1
704 0226 1
705 0227 1
706 0228 1
707 0229 1
708 0220 1
709 0221 1
710 0222 1
711 0223 1
712 0224 1
713 0225 1
714 0226 1
715 0227 1
716 0228 1
717 0229 1
718 0220 1
719 0221 1
720 0222 1
721 0223 1
722 0224 1
723 0225 1
724 0226 1
725 0227 1
726 0228 1
727 0229 1
728 0220 1
729 0221 1
730 0222 1
731 0223 1
732 0224 1
733 0225 1
734 0226 1
735 0227 1
736 0228 1
737 0229 1
738 0220 1
739 0221 1
740 0222 1
741 0223 1
742 0224 1
743 0225 1
744 0226 1
745 0227 1
746 0228 1
747 0229 1
748 0220 1
749 0221 1
750 0222 1
751 0223 1
752 0224 1
753 0225 1
754 0226 1
755 0227 1
756 0228 1
757 0229 1
758 0220 1
759 0221 1
760 0222 1
761 0223 1
762 0224 1
763 0225 1
764 0226 1
765 0227 1
766 0228 1
767 0229 1
768 0220 1
769 0221 1
770 0222 1
771 0223 1
772 0224 1
773 0225 1
774 0226 1
775 0227 1
776 0228 1
777 0229 1
778 0220 1
779 0221 1
780 0222 1
781 0223 1
782 0224 1
783 0225 1
784 0226 1
785 0227 1
786 0228 1
787 0229 1
788 0220 1
789 0221 1
790 0222 1
791 0223 1
792 0224 1
793 0225 1
794 0226 1
795 0227 1
796 0228 1
797 0229 1
798 0220 1
799 0221 1
800 0222 1
801 0223 1
802 0224 1
803 0225 1
804 0226 1
805 0227 1
806 0228 1
807 0229 1
808 0220 1
809 0221 1
810 0222 1
811 0223 1
812 0224 1
813 0225 1
814 0226 1
815 0227 1
816 0228 1
817 0229 1
818 0220 1
819 0221 1
820 0222 1
821 0223 1
822 0224 1
823 0225 1
824 0226 1
825 0227 1
826 0228 1
827 0229 1
828 0220 1
829 0221 1
830 0222 1
831 0223 1
832 0224 1
833 0225 1
834 0226 1
835 0227 1
836 0228 1
837 0229 1
838 0220 1
839 0221 1
840 0222 1
841 0223 1
842 0224 1
843 0225 1
844 0226 1
845 0227 1
846 0228 1
847 0229 1
848 0220 1
849 0221 1
850 0222 1
851 0223 1
852 0224 1
853 0225 1
854 0226 1
855 0227 1
856 0228 1
857 0229 1
858 0220 1
859 0221 1
860 0222 1
861 0223 1
862 0224 1
863 0225 1
864 0226 1
865 0227 1
866 0228 1
867 0229 1
868 0220 1
869 0221 1
870 0222 1
871 0223 1
872 0224 1
873 0225 1
874 0226 1
875 0227 1
876 0228 1
877 0229 1
878 0220 1
879 0221 1
880 0222 1
881 0223 1
882 0224 1
883 0225 1
884 0226 1
885 0227 1
886 0228 1
887 0229 1
888 0220 1
889 0221 1
890 0222 1
891 0223 1
892 0224 1
893 0225 1
894 0226 1
895 0227 1
896 0228 1
897 0229 1
898 0220 1
899 0221 1
900 0222 1
901 0223 1
902 0224 1
903 0225 1
904 0226 1
905 0227 1
906 0228 1
907 0229 1
908 0220 1
909 0221 1
910 0222 1
911 0223 1
912 0224 1
913 0225 1
914 0226 1
915 0227 1
916 0228 1
917 0229 1
918 0220 1
919 0221 1
920 0222 1
921 0223 1
922 0224 1
923 0225 1
924 0226 1
925 0227 1
926 0228 1
927 0229 1
928 0220 1
929 0221 1
930 0222 1
931 0223 1
932 0224 1
933 0225 1
934 0226 1
935 0227 1
936 0228 1
937 0229 1
938 0220 1
939 0221 1
940 0222 1
941 0223 1
942 0224 1
943 0225 1
944 0226 1
945 0227 1
946 0228 1
947 0229 1
948 0220 1
949 0221 1
950 0222 1
951 0223 1
952 0224 1
953 0225 1
954 0226 1
955 0227 1
956 0228 1
957 0229 1
958 0220 1
959 0221 1
960 0222 1
961 0223 1
962 0224 1
963 0225 1
964 0226 1
965 0227 1
966 0228 1
967 0229 1
968 0220 1
969 0221 1
970 0222 1
971 0223 1
972 0224 1
973 0225 1
974 0226 1
975 0227 1
976 0228 1
977 0229 1
978 0220 1
979 0221 1
980 0222 1
981 0223 1
982 0224 1
983 0225 1
984 0226 1
985 0227 1
986 0228 1
987 0229 1
988 0220 1
989 0221 1
990 0222 1
991 0223 1
992 0224 1
993 0225 1
994 0226 1
995 0227 1
996 0228 1
997 0229 1
998 0220 1
999 0221 1
1000 0222 1
1001 0223 1
1002 0224 1
1003 0225 1
1004 0226 1
1005 0227 1
1006 0228 1
1007 0229 1
1008 0220 1
1009 0221 1
1010 0222 1
1011 0223 1
1012 0224 1
1013 0225 1
1014 0226 1
1015 0227 1
1016 0228 1
1017 0229 1
1018 0220 1
1019 0221 1
1020 0222 1
1021 0223 1
1022 0224 1
1023 0225 1
1024 0226 1
1025 0227 1
1026 0228 1
1027 0229 1
1028 0220 1
1029 0221 1
1030 0222 1
1031 0223 1
1032 0224 1
1033 0225 1
1034 0226 1
1035 0227 1
1036 0228 1
1037 0229 1
1038 0220 1
1039 0221 1
1040 0222 1
1041 0223 1
1042 0224 1
1043 0225 1
1044 0226 1
1045 0227 1
1046 0228 1
1047 0229 1
1048 0220 1
1049 0221 1
1050 0222 1
1051 0223 1
1052 0224 1
1053 0225 1
1054 0226 1
1055 0227 1
1056 0228 1
1057 0229 1
1058 0220 1
1059 0221 1
1060 0222 1
1061 0223 1
1062 0224 1
1063 0225 1
1064 0226 1
1065 0227 1
1066 0228 1
1067 0229 1
1068 0220 1
1069 0221 1
1070 0222 1
1071 0223 1
1072 0224 1
1073 0225 1
1074 0226 1
1075 0227 1
1076 0228 1
1077 0229 1
1078 0220 1
1079 0221 1
1080 0222 1
1081 0223 1
1082 0224 1
1083 0225 1
1084 0226 1
1085 0227 1
1086 0228 1
1087 0229 1
1088 0220 1
1089 0221 1
1090 0222 1
1091 0223 1
1092 0224 1
1093 0225 1
1094 0226 1
1095 0227 1
1096 0228 1
1097 0229 1
1098 0220 1
1099 0221 1
1100 0222 1
1101 0223 1
1102 0224 1
1103 0225 1
1104 0226 1
1105 0227 1
110
```

163 0225 1
164 0226 1
165 0227 1
166 0228 1
167 0229 1
168 0230 1
169 0231 1
170 0232 1
171 0233 1
172 0234 1
173 0235 1
174 0236 1
175 0237 1
176 0238 1
177 0239 1
178 0240 1
179 0241 1
180 0242 1
181 0243 1
182 0244 1
183 0245 1
184 0246 1
185 0247 1
186 0248 1
187 0249 1
188 0250 1
189 0251 1
190 0252 1
191 0253 1
192 0254 1
193 0255 1
194 0256 1
195 0257 1
196 0258 1
197 0259 1
198 0260 1
199 0261 1
200 0262 1
201 0263 1
202 0264 1
203 0265 1
204 0266 1
205 0267 1
206 0268 1
207 0269 1
208 0270 1
209 0271 1
210 0272 1
211 0273 1
212 0274 1
213 0275 1
214 0276 1
215 0277 1

0225 1
0226 1
0227 1
0228 1
0229 1
0230 1
0231 1
0232 1
0233 1
0234 1
0235 1
0236 1
0237 1
0238 1
0239 1
0240 1
0241 1
0242 1
0243 1
0244 1
0245 1
0246 1
0247 1
0248 1
0249 1
0250 1
0251 1
0252 1
0253 1
0254 1
0255 1
0256 1
0257 1
0258 1
0259 1
0260 1
0261 1
0262 1
0263 1
0264 1
0265 1
0266 1
0267 1
0268 1
0269 1
0270 1
0271 1
0272 1
0273 1
0274 1
0275 1
0276 1
0277 1

RPG\$W_CTX_SPACEB number of lines to space before printing.
RPG\$W_CTX_SPACEA number of lines to space after printing.
RPG\$W_CTX_SKIPB line number to skip to before printing.
RPG\$W_CTX_SKIPA line number to skip to after printing.
RPG\$W_CTX_PFLAGS flags for print control:
RPG\$V_CTX_FIRST TRUE before first write to the file to ensure that values get initialized and that the "first page" forms positioning takes place, if requested, on the first write.
RPG\$V_CTX_1PFORMS TRUE when "first page" forms positioning has been requested.
RPG\$V_CTX_OVLINE TRUE when this is an overflow line.
RPG\$W_CTX_LINE specifies the line number at which the device is positioned within the current page body.
RPG\$W_CTX_FL specifies the number of lines in the page body; i.e., it specifies the number of lines on the logical page that can be written.
RPG\$W_CTX_OL specifies the line number of overflow line.
RPG\$A_CTX_OV_ND specifies the address of the overflow indicator for this file.
RABSL_RHB is the address of the two byte control area to contain the print file information. The first byte is the "prefix" area, and the second byte is the "postfix" area, specifying the number of lines to advance before and after the record, respectively.

IMPLICIT OUTPUTS:
NONE

ROUTINE VALUE:
RMS status returned by the PUT operation or RPG\$_EXTINDOFF.

SIDE EFFECTS:
A PUT to the linage file is performed.

```
217 0278 2 BEGIN
218 0279 2
219 0280 2 LITERAL
220 0281 2     SET_ON = 1
221 0282 2     SET_OFF = 0
222 0283 2     SET_OFF_OVERFLOW = %X'FFFEFEFE',
223 0284 2
224 0285 2
225 0286 2     SET_ON_OVERFLOW = %X'00010101';
226 0287 2
227 0288 2
228 0289 2
229 0290 2
230 0291 2
231 0292 2 LOCAL
232 0293 2     ADV_LINES,
233 0294 2     LINE_FLAG: WORD,
234 0295 2
235 0296 2
236 0297 2     RET_STATUS,
237 0298 2     RHB: REF BLOCK[,BYTE];
238 0299 2
239 0300 2
240 0301 2     FCB = RAB : REF BLOCK [,BYTE];      ! File context block
241 0302 2
242 0303 2
243 0304 2     BUILTIN
244 0305 2     TESTBITS;
245 0306 2
246 0307 2
247 0308 2
248 0309 2
249 0310 2
250 0311 2
251 0312 2
252 0313 2     IF .RAB[RABSW_ISI] EQL 0
253 0314 2     THEN
254 0315 2     RETURN RPG$EXTINOFF;
255 0316 2
256 0317 2
257 0318 2     LINE_FLAG = .FCB[RPG$W_CTX_LINE];
258 0319 2     RHB = .RAB[RABSL_RHB];
259 0320 2     RHB[PREFIX] = 0;
260 0321 2     RHB[POSTFIX] = 0;
261 0322 2
262 0323 2
263 0324 2
264 0325 2
265 0326 2
266 0327 2
267 0328 3     IF .FCB[RPG$W_CTX_SKIPB] GTR 0
268 0329 3     THEN
269 0330 3     BEGIN
270 0331 3     ! Skip before
271 0332 3     SKIP BEFORE indicated
272 0333 3     ADV_LINES = .FCB[RPG$W_CTX_SKIPB] - .FCB[RPG$W_CTX_LINE];
273 0334 3     IF .ADV_LINES NEQ 0
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
```

```
274 0335 3      THEN
275 0336 4        BEGIN          ! New line
276 0337 4
277 0338 4        FCB[RPG$W_CTX_LINE] = .FCB[RPG$W_CTX_SKIPB];      ! Update current line
278 0339 4        IF .ADV_LINES LSS 0
279 0340 4        THEN
280 0341 5          BEGIN
281 0342 5
282 0343 5          ! SKIP BEFORE will cause advance to a new page
283 0344 5
284 0345 5          RHB[PREFIX] = .FCB[RPG$W_CTX_FL] + .ADV_LINES;      ! Set prefix in control area
285 0346 5          LINE_FLAG = 0;      ! Flag reset for new page
286 0347 5          FCB[RPG$V_CTX_OVPEND] = SET_OFF;      ! 1-003 Flag reset for new page
287 0348 5          IF .FCB[RPG$V_CTX_OVLINE] NEQ SET_ON
288 0349 5          THEN
289 0350 5            .FCB[RPG$A_CTX_OVIND] = ..FCB[RPG$A_CTX_OVIND] AND SET_OFF_OVERFLOW;      ! Set off the overflow indicator
290 0351 5            ! Set off the overflow indicator
291 0352 5
292 0353 5        END
293 0354 4      ELSE
294 0355 4
295 0356 4          ! SKIP TO line will be on the same page
296 0357 4
297 0358 4          RHB[PREFIX] = .ADV_LINES;      ! Set prefix in control area
298 0359 4
299 0360 4
300 0361 3      END;          ! New line
301 0362 3
302 0363 2      END;          ! Skip before
303 0364 2
304 0365 2      IF .FCB[RPG$W_CTX_SPACEB] GTR 0
305 0366 2      THEN
306 0367 3        BEGIN
307 0368 3
308 0369 3          ! SPACE BEFORE indicated
309 0370 3
310 0371 3        FCB[RPG$W_CTX_LINE] = .FCB[RPG$W_CTX_LINE] + .FCB[RPG$W_CTX_SPACEB];      ! Update current line
311 0372 3
312 0373 3        RHB[PREFIX] = .RHB[PREFIX] + .FCB[RPG$W_CTX_SPACEB];      ! Adjust prefix in control area
313 0374 3
314 0375 3      END
315 0376 2      ELSE
316 0377 2
317 0378 2          ! If the skip caused no advance, then we are going to print on
318 0379 2          ! the same line as the previous PUT, so we need the specify CR
319 0380 2          ! in the prefix area to get overprinting.
320 0381 2
321 0382 2      IF .RHB[PREFIX] EQL 0
322 0383 2      THEN
323 0384 2        RHB[PREFIX] = XX'8D';      ! 1-003
324 0385 2
325 0386 2
326 0387 2      ! Check for line being printed on or past the overflow line
327 0388 2
328 0389 2      IF .FCB[RPG$W_CTX_LINE] GEQ .FCB[RPG$W_CTX_OL]
329 0390 2      THEN
330 0391 2        IF (.LINE_FLAG LSS .RAB[RPG$W_CTX_OL]) OR      ! First time on this page?
```

```
331      0392 3      (.FCB[RPG$V_CTX_OVPEND] EQL SET_ON)           ! 1-003 Was an overflow pending?
332      0393 2
333      0394 2      THEN
334      0395 3      BEGIN
335      0396 3      .FCB[RPG$A_CTX_OVIND] = ..FCB[RPG$A_CTX_OVIND] OR SET_ON_OVERFLOW;
336      0397 3
337      0398 3      FCB[RPG$V_CTX_OVPEND] = SET_OFF;           ! Yes set on the overflow indicator
338      0399 3
339      0400 2      END;
340      0401 2
341      0402 2
342      0403 2      | Check for current line being on new page
343      0404 2
344      0405 2      IF .FCB[RPG$W_CTX_LINE] GTR .FCB[RPG$W_CTX_FL]
345      0406 2      THEN
346      0407 2      FCB[RPG$W_CTX_LINE] = .FCB[RPG$W_CTX_LINE] - .FCB[RPG$W_CTX_FL]; ! Adjust current line to reflect
347      0408 2
348      0409 2
349      0410 2
350      0411 2      | Process skipping and spacing after the print
351      0412 2
352      0413 2      IF .FCB[RPG$W_CTX_SKIPA] GTR 0
353      0414 2      THEN
354      0415 3      BEGIN           ! Skip after
355      0416 3
356      0417 3      | SKIP AFTER indicated
357      0418 3
358      0419 3      ADV_LINES = .FCB[RPG$W_CTX_SKIPA] - .FCB[RPG$W_CTX_LINE];           ! Number of lines to advance
359      0420 3      IF .ADV_LINES NEQ 0           ! Make sure SKIP TO line
360      0421 3
361      0422 3      THEN
362      0423 4      BEGIN           ! New line
363      0424 4      FCB[RPG$W_CTX_LINE] = .FCB[RPG$W_CTX_SKIPA];           ! Update current line
364      0425 4      IF .ADV_LINES LSS 0
365      0426 4      THEN
366      0427 5      BEGIN
367      0428 5
368      0429 5      | SKIP AFTER will cause advance to a new page
369      0430 5
370      0431 5      RHB[POSTFIX] = .FCB[RPG$W_CTX_FL] + .ADV_LINES;           ! Set postfix in control area
371      0432 5      LINE FLAG = 0;           ! Reset flag for new page
372      0433 5
373      0434 5      IF .FCB[RPG$V_CTX_OVLINE] NEQ SET_ON
374      0435 5      .FCB[RPG$A_CTX_OVIND] = ..FCB[RPG$A_CTX_OVIND] AND SET_OFF_OVERFLOW;
375      0436 5
376      0437 5
377      0438 5      ELSE
378      0439 4
379      0440 4
380      0441 4      | SKIP AFTER line will be on the same page
381      0442 4
382      0443 4      RHB[POSTFIX] = .ADV_LINES;           ! Set postfix in control area
383      0444 4
384      0445 3
385      0446 3
386      0447 2      END;           ! New line
387      0448 2
388
```

```
388 0449 2 IF .FCB[RPGSW_CTX_SPACEA] GTR 0
389 0450 2 THEN
390 0451 3 BEGIN
391 0452 3
392 0453 3 | SPACE AFTER indicated
393 0454 3
394 0455 3 FCB[RPGSW_CTX_LINE] = .FCB[RPGSW_CTX_LINE] + .FCB[RPGSW_CTX_SPACEA];
395 0456 3
396 0457 3 RHB[POSTFIX] = .RHB[POSTFIX] + .FCB[RPGSW_CTX_SPACEA]; | Update current line
397 0458 3 | Adjust postfix in control area
398 0459 2
399 0460 2
400 0461 2
401 0462 2 | Check for overflow line being passed by space or skip
402 0463 2
403 0464 2 IF (.FCB[RPGSW_CTX_LINE] GTR .FCB[RPGSW_CTX_DL]) AND | 1-003 OL passed during skip?
404 0465 3 (.LINE_FLAG LSS .FCB[RPGSW_CTX_DL]) | First time on this page?
405 0466 2 THEN
406 0467 2 .FCB[RPGSA_CTX_OVIND] = ..FCB[RPGSA_CTX_OVIND] OR SET_ON_OVERFLOW | Yes, set on the overflow indicator
407 0468 2
408 0469 2
409 0470 2 ELSE IF (.FCB[RPGSW_CTX_LINE] EQL .FCB[RPGSW_CTX_DL]) AND | 1-003 OL reached during space or s
410 0471 3 (.LINE_FLAG LSS .FCB[RPGSW_CTX_DL]) | 1-003 First time on this page?
411 0472 2 THEN
412 0473 2 FCB[RPGSV_CTX_OVPEND] = SET_ON; | 1-003 Flag that overflow is pending
413 0474 2
414 0475 2
415 0476 2 | Check for current line being on a new page
416 0477 2
417 0478 2 IF .FCB[RPGSW_CTX_LINE] GTR .FCB[RPGSW_CTX_FL]
418 0479 2 THEN
419 0480 2 FCB[RPGSW_CTX_LINE] = .FCB[RPGSW_CTX_LINE] - .FCB[RPGSW_CTX_FL]; | Adjust current line to reflect
420 0481 2 | new page
421 0482 2
422 0483 2
423 0484 2 | It is necessary to special-case the first WRITE on the first logical
424 0485 2 | page after a file has been OPENed so that 'first page' forms
425 0486 2 | positioning can be done.
426 0487 2
427 0488 2 IF TESTBITS(FCB[RPGSV_CTX_FIRST])
428 0489 2 THEN
429 0490 2 IF .FCB[RPGSV_CTX_1PFORMS]
430 0491 2 THEN
431 0492 3 BEGIN | First page forms positioning
432 0493 3
433 0494 3 LOCAL
434 0495 3 GET_STATUS, | Return status from LIB$GET_COMMAND
435 0496 3 PROMPT_DESC: BLOCK[8,BYTE], | Local descriptor for prompt
436 0497 3 RESP_DESC: BLOCK[8,BYTE], | Local descriptor for response
437 0498 3 RESP_BUF: VECTOR[10,BYTE]; | Buffer for response
438 0499 3
439 0500 3 LITERAL
440 0501 3 TRUE = 1,
441 0502 3 MIN_RESP_LEN = %CHARCOUNT('xxx'); | Minimum acceptable length of
442 0503 3 | response to LIB$GET_COMMAND
443 0504 3
444 0505 3
```

```
445 0506 3      LABEL
446 0507 3      OUTER_LOOP;
447 0508 3
448 0509 3
449 0510 3
450 0511 3      | NOTE - PROMPT must come directly before RET for the prompt
451 0512 3      | string length to be calculated correctly
452 0513 3
453 0514 3      | PROMPT = UPLIT (' Is forms positioning correct? Yes, type CONTINUE, No, type RETRY: '),
454 0515 3      | RET = UPLIT ('RET'),
455 0516 3      | CON = UPLIT ('CON');
456 0517 3
457 0518 3
458 0519 3      | 'First page' forms positioning indicated
459 0520 3
460 0521 3      | PROMPT_DESC[DSC$W_LENGTH] = CH$DIFF (RET, PROMPT);
461 0522 3      | PROMPT_DESC[DSC$B_CLASS] = DSC$K_CLASS_S;
462 0523 3      | PROMPT_DESC[DSC$B_DTYPE] = DSC$K_DTYPE_T;
463 0524 3      | PROMPT_DESC[DSC$A_POINTER] = PROMPT;
464 0525 3      | RESP_DESC[DSC$W_LENGTH] = %ALLOCATION (RESP_BUF);
465 0526 3      | RESP_DESC[DSC$B_CLASS] = DSC$K_CLASS_S;
466 0527 3      | RESP_DESC[DSC$B_DTYPE] = DSC$K_DTYPE_T;
467 0528 3      | RESP_DESC[DSC$A_POINTER] = RESP_BUF;
468 0529 3
469 0530 4 OUTER_LOOP: BEGIN
470 0531 4
471 0532 4      WHILE TRUE DO
472 0533 5      BEGIN          ! Retry loop
473 0534 5
474 0535 5      | PUT the record
475 0536 5
476 0537 5      | RET STATUS = $PUT(RAB = .RAB);          ! Put out the record
477 0538 6      | IF NOT (.RET_STATUS)
478 0539 5      | THEN
479 0540 6      | BEGIN
480 0541 6
481 0542 6      | Error on PUT, return
482 0543 6
483 0544 6      | FCB[RPG$V_CTX_FIRST] = SET_ON;          ! Reset FIRST bit
484 0545 6      | RETURN .RET_STATUS;
485 0546 6
486 0547 5
487 0548 5
488 0549 5      | Issue a message to SYSSCOMMAND to ask the user
489 0550 5      | whether forms are positioned correctly.
490 0551 5      | If response is neither RET(RY) or CON(TINUE),
491 0552 5      | prompt again.
492 0553 5      | If response is RETRY, go thru outer loop again.
493 0554 5
494 0555 5      WHILE TRUE DO
495 0556 6      BEGIN
496 0557 6
497 0558 6      DO          ! Prompt for response until user types
498 0559 6          | ! RET(RY) or CON(TINUE)
499 0560 6
500 0561 6      GET STATUS = LIB$GET_COMMAND(RESP_DESC, PROMPT_DESC)
501 0562 6      UNTIL .GET_STATUS;
```

```

502 0563 6      STR$UPCASE (RESP_DESC, RESP_DESC);
503 0564 6      IF CH$EQ (MIN_RESP_LEN, RESP_BUF, MIN_RESP_LEN, CON)
504 0565 6      THEN
505 0566 6      LEAVE OUTER LOOP;
506 0567 6      IF CH$EQ (MIN_RESP_LEN, RESP_BUF, MIN_RESP_LEN, RET)
507 0568 6      THEN
508 0569 6      EXITLOOP;
509 0570 6
510 0571 5      END;
511 0572 5
512 0573 4      END;          ! Retry loop
513 0574 4
514 0575 3      END;          ! Outer loop
515 0576 3      RETURN .RET_STATUS;          ! Return status from PUT
516 0577 3
517 0578 2      END;          ! First page forms positioning
518 0579 2
519 0580 2
520 0581 2      ! When not special-casing, will get here.
521 0582 2
522 0583 2      RETURN SPUT(RAB = .RAB);          ! PUT out the record and
523 0584 2          ! return the RMS status
524 0585 2
525 0586 1      END;

```

```

.TITLE RPG$PRINT Support output to RPG PRINTER files
.IDENT \1-003\

.PSECT _RPG$CODE,NOWRT, SHR, PIC,2

74 69 73 6F 70 20 73 6D 72 6F 66 20 73 49 20 00000 P.AAA: .ASCII \ Is forms positioning correct? Yes, type\ :
3F 74 63 65 72 72 6F 63 20 67 6E 69 6E 6F 69 0000F
20 2C 6F 4E 20 2C 45 55 4E 49 54 4E 4F 43 20 00028 .ASCII \ CONTINUE, No, type RETRY: \<0>
00 20 3A 59 52 54 45 52 20 65 70 79 74 00037
          00 54 45 52 00044 P.AAB: .ASCII \RET\<0>
          00 4E 4F 43 00048 P.AAC: .ASCII \CON\<0>

PROMPT=          P.AAA
RET=            P.AAB
CON=            P.AAC
                .EXTRN LIB$GET_COMMAND
                .EXTRN STR$UPCASE, RPG$_EXTINOFF
                .EXTRN SY$PUT

                .ENTRY RPG$PRINT, Save R2,R3,R4,R5,R6,R7
                .0169
                MOVAB 00FC 00000
                57 0000C000G 00 9E 00002
                5E 1C C2 00009
                54 04 AC D0 0000C
                02 A4 B5 00010
                08 12 00013
                .0311
                50 00000000G 8F D0 00015
                04 0001C
                .0313
                50 EE A4 9E 0001D 1$: RET
                55 60 B0 00021
                51 2C A4 D0 00024
                .0318
                MOVAB -18(R4), R0
                MOVW (R0), LINE_FLAG
                MOVL 44(R4), RHB
                .0319

```


		10	11	000EE		BRB	13\$		0470	
		60	B1	000FO	12\$:	CMPW	(R0), -16(R4)			
		0A	12	000F4		BNEQ	13\$		0471	
		55	B1	000F6		CMPW	LINE_FLAG, -16(R4)			
		04	1E	000FA		BGEQU	13\$		0473	
		EC	A4	08	88	000FC	BISB2	#8 -20(R4)		
		F2	A4	60	B1	00100	13\$:	CMPW	(R0), -14(R4)	
				04	1B	00104	BLEQU	14\$	0478	
			60	A2	00106		SUBW2	-14(R4), (R0)		
		61	EC	00	E5	0010A	14\$:	BBCC	#0, -20(R4), 18\$	
			EC	A4	01	E1	0010F	BBC	#1 -20(R4), 18\$	
		66	14	AE	010E0044	8F	D0	00114	MOVL	#17694788, PROMPT DESC
			18	AE	FE94	CF	9E	0011C	MOVAB	PROMPT DESC+4
			OC	AE	010E000A	8F	D0	00122	MOVL	#17694730, RESP DESC
			10	AE		6E	9E	0012A	MOVAB	RESP_BUF, RESP_DESC
					54	DD	OC12E	15\$:	PUSHL	R4
					01	FB	00130		CALLS	#1, SYSSPUT
					55	50	D0	00133	MOVL	RO, RET STATUS
					06	55	E8	00136	BLBS	RET_STATUS, 16\$
					EC	A4	01	88	BISB2	#1 -20(R4)
					67	32	11	0013D	BRB	17\$
						14	AE	9F	PUSHAB	PROMPT DESC
						10	AE	9F	PUSHAB	RESP DESC
							00000000G	00	CALLS	#2, LIB\$GET COMMAND
							56	50	MOVL	RO, GET STATUS
							ED	56	BLBC	GET STATUS, 16\$
								OC	PUSHAB	RESP DESC
								10	PUSHAB	RESP DESC
									CALLS	#2, STR\$UPCASE
									CMPC3	#3, RESP_BUF, CON
									BEQL	17\$
									CMPC3	#3, RESP_BUF, RET
									BNEQ	16\$
									BRB	15\$
									MOVL	RET_STATUS, RO
									RET	
									PUSHL	R4
									CALLS	#1, SYSSPUT
									RET	

; Routine Size: 379 bytes. Routine Base: _RPG\$CODE + 004C

```
527 0587 1 %SBTTL 'RPG$TERM_PRINT - Finish logical page'
528 0588 1 GLOBAL ROUTINE RPG$TERM_PRINT(
529 0589 1 )=          RAB:  REF $RAB_DECL ! RAB of file to be printed
530 0590 1
531 0591 1
532 0592 1 ++
533 0593 1
534 0594 1 FUNCTIONAL DESCRIPTION:
535 0595 1
536 0596 1 This routine is called to advance the number of lines needed to
537 0597 1 finish out the logical page before the actual CLOSE is done.
538 0598 1
539 0599 1
540 0600 1 CALLING SEQUENCE:
541 0601 1
542 0602 1     return_status.wlc.v = RPG$TERM_PRINT (rab.rr.r)
543 0603 1
544 0604 1 FORMAL PARAMETERS:
545 0605 1
546 0606 1     rab           address of the RAB of the file to be
547 0607 1           printed.
548 0608 1
549 0609 1 IMPLICIT INPUTS:
550 0610 1
551 0611 1     RPGSW_CTX_FL      specifies the number of lines in the page body;
552 0612 1           i.e., it specifies the number of lines on the
553 0613 1           logical page that can be written.
554 0614 1
555 0615 1     RPGSW_CTX_LINE    specifies the line number at which the device is
556 0616 1           positioned within the current page body.
557 0617 1
558 0618 1
559 0619 1 IMPLICIT OUTPUTS:
560 0620 1
561 0621 1     A PUT to the linage file is performed
562 0622 1
563 0623 1 ROUTINE VALUE:
564 0624 1
565 0625 1     RMS status returned by the PUT operation or SSS_NORMAL if
566 0626 1           nothing needs to be done by this routine.
567 0627 1
568 0628 1
569 0629 1
570 0630 1
571 0631 1
572 0632 1 SIDE EFFECTS:
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
```

```
574 0633 2 BEGIN
575 0634 2
576 0635 2 LITERAL
577 0636 2     SET_ON = 1;
578 0637 2
579 0638 2 LOCAL
580 0639 2     RHB : REF BLOCK [,BYTE];           ! Record header block
581 0640 2
582 0641 2 BIND
583 0642 2     FCB = RAB : REF BLOCK [,BYTE];       ! File context block
584 0643 2
585 0644 2
586 0645 2     RPG$TERM_PRINT should not cause access violations. Since it WILL
587 0646 2     be called before the associated SYSSCLOSE, the RAB may be invalid.
588 0647 2     Validate the RAB by checking that RAB$W_ISI is non-zero.
589 0648 2
590 0649 2 IF .RAB[RAB$W_ISI] EQL 0
591 0650 2 THEN
592 0651 2     RETURN RPG$_EXTINDOFF;
593 0652 2
594 0653 2
595 0654 2     If no WRITE has ever been done for this file, then no adjustment
596 0655 2     need be done at CLOSE time. Note that the flag bit is checked
597 0656 2     but not cleared; if it is set, we will not be doing a WRITE either.
598 0657 2
599 0658 2 IF .FCB[RPG$V_CTX_FIRST] EQL SET_ON
600 0659 2 THEN
601 0660 2     RETURN SSS_NORMAL;
602 0661 2
603 0662 2
604 0663 2     Figure out how many lines left to fill out the page
605 0664 2
606 0665 2     RHB = .RAB[RAB$L_RHB];
607 0666 2     RHBC[PREFIX] = .FCB[RPG$W_CTX_FL] - .FCB[RPG$W_CTX_LINE] + 1;
608 0667 2
609 0668 2
610 0669 2     Make sure that there is something to advance.
611 0670 2
612 0671 2 IF .RHB[PREFIX] EQL 0
613 0672 2 THEN
614 0673 2     RETURN SSS_NORMAL;
615 0674 2
616 0675 2
617 0676 2     The actual WRITE is done by PUTing a record of 0 length with appropriate
618 0677 2     advance in the PRN control fields.
619 0678 2
620 0679 2     RAB[RAB$W_RSZ] = 0;
621 0680 2     RHBC[POSTFIX] = 0;
622 0681 2
623 0682 2     RETURN $PUT(RAB = .RAB);
624 0683 2
625 0684 1     END;
```

RPGSPRINT
1-003

Support output to RPG PRINTER files
RPG\$TERM_PRINT - Finish logical page

C 4
16-Sep-1984 02:18:04 VAX-11 Bliss-32 V4.0-742
14-Sep-1984 13:04:24 [RPGRTL.SRC]RPGPRINT.B32;1

Page 15
(6)

RPG
1-0

; Routine Size: 64 bytes, Routine Base: _RPG\$CODE + 01C7

: 626 0685 1
: 627 0686 0 END FLUDGM

PSECT SUMMARY

Name	Bytes	Attributes
RPG\$CODE	519	NOVEC,NOWRT, RD , EXE, SHR, LCL, REL, CON, PIC,ALIGN(2)

Library Statistics

File	----- Symbols -----			Pages Mapped	Processing Time
	Total	Loaded	Percent		
-\$255\$DUA28:[SYSLIB]STARLET.L32;1	9776	16	0	581	00:01.0
-\$255\$DUA28:[RPGRTL.OBJ]RPGLIB.L32;1	54	12	22	9	00:00.1

COMMAND QUALIFIERS

RPG\$PRINT
1-003

Support output to RPG PRINTER files
RPG\$TERM_PRINT - Finish logical page

D 4
16-Sep-1984 02:18:04
14-Sep-1984 13:04:24

VAX-11 Bliss-32 V4.0-742
[RPGRTL.SRC]RPGPRINT.B32;1

Page 16
(6)

: BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/NOTRACE/LIS=LISS:RPGPRINT/OBJ=OBJ\$:RPGPRINT MSRC\$:RPGPRINT/UPDATE=(ENH\$:RPGPRINT)

: Size: 443 code + 76 data bytes
: Run Time: 00:13.7
: Elapsed Time: 00:46.1
: Lines/CPU Min: 3000
: Lexemes/CPU-Min: 21161
: Memory Used: 189 pages
: Compilation Complete

0332 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

RPGMSGTXT
LIS

DTE DF03
MAP

RPGMOVE3
LIS

RPGSORT
LIS

RPGOPEN
LIS

RTPAD

CTDRIVER
MAP

MAP

RTPAD
MAP

RTPADMACS
MAP

RPGMSGPTR
LIS

RPGVECTOR
LIS

RTDEF
SDL

DTE DF03
MAP

RPGPRINT
LIS

RPGUPDATE
LIS

CTDRIVER
LIS